Evaluating the Performance on ID/Loc Mapping

Hong Zhang
Department of Electronic Engineering
Tsinghua University
Beijing, 100084, P.R.China
Email: neilzh@ns.6test.edu.cn

Maoke Chen
Network Research Center
Tsinghua University
Beijing, 100084, P.R.China
Email: cmk@ns.6test.edu.cn

Yuncheng Zhu
Department of Electronic Engineering
Tsinghua University
Beijing, 100084, P.R.China
Email:zhu@ns.6test.edu.cn

***Abstract*—Challenges of routing scalability has attracted many research efforts, represented by the works of splitting identifier and locator semantics of IP addresses. A group of identifier-locator (ID/Loc) split approaches is commonly featured with a mapping query service system, independent of the routing infrastructure. This is a significant change of the Internet routing architecture that deserves comprehensive analysis and quantitative evaluations. Focusing on the mostly concerned performance issues, we present a canonical model as the typical case where the mapping query is executed for routers performing the ID/Loc mappings, and then evaluate the behaviors of caching, query retrieval and queueing introduced by the query latency. According to the results, a well-defined mapping service is able to handle the traffic volume that a current big provider may experience. Furthermore, we also suggest an end system modification to get better performance in the age of ID/Loc having been split.**

I. Introduction

In recent years, scaling problems faced by the Internet routing architecture are attracting more and more attentions. The Forwarding Information Base (FIB) of the Default Free Zone (DFZ) keeps growing at an alarming rate. It has been suggested that the overloaded role of IP address as both end host identifier and routing locator is one of the root causes of this scalability problem [1]. Separating the single numbering space of IP address into two for identifier and locator can bring many benefits on routing table reduction in DFZ, traffic engineering in multi-homing and so on. These benefits of the separation have been already evaluated [2].

In order to realize this separation, several novel architectures are proposed, such as LISP [3], LISP2 [4], LISP-CONS [5], LISP-ALT [6], NERD [7], and APT [8]. Many of them contain a query system for the identifier-locator (ID/Loc) mappings, which is a service independent of the routing infrastructure. This is a significant change of current Internet's routing architecture. It has become a common understanding that the Internet must be changed in the future [9], though it is still necessary to quantitatively evaluate the impact of introducing such a query service system as a necessary ingredient for end-to-end connectivity [10].

It is a big challenge for the end-to-end performance that forwarding a packet relies on efficiently querying the proper mapping entry out. Packets will be queued or even dropped if a gateway router does not have the proper identifier-locator correspondence when a packet arrives. Locally caching mapping entries is a reasonable design to accelerate packet delivery but globally performing the query cannot be completely avoided. Furthermore, latency cause by query will also change the queueing behavior of routers.

In this paper, with the purpose of understanding the above challenges, we present a canonical model of the mapping-routing facility in the hypothetic Internet where identifier and locator have been split from IP address and the mapping service system has been deployed. A unit of the facility in the model consists of a local cache, a query substrate and a packet queueing machine. Real traffic data are fed into the model to simulate the behavior of the mapping facility.

The remaining parts of the paper are organized as follows. In Section II, we present a canonical model of the facility that may perform ID/Loc mapping through local cache and/or global query. Then, Section III evaluates the performance of the canonical model in terms of the size of local cache, the load of query system and the packet queueing behavior in routers. According to the experiment results, we discuss the performance of the model in Section IV and give a suggestion on modifying TCP behaviors of hosts, in the future Internet where Id/Loc split has been implemented.

II. The Query-Cache-Queue Model

LISP2 [4], LISP-CONS [5] and NERD [7] and 6Core [11] have a common overall structure consisting of three basic components: the query system, the local cache and the queue on the gateway router. This canonical model is depicted by Fig. 1.

A. Query System

In the canonical model, the query system is a mapping index facility independent from the routing infrastructure. The possible implementation of such a query system could be an sub-zone in ever existing DNS (Domain Name System) [4], a dedicated, decentralized database [11], or a peer-to-peer index facility like DHT (distributed hash table). No matter how the query system is implemented, it faces a common challenge of latency in the query response. Therefore, we can define the abstract query system as a basic component in the canonical model and focus our further analysis on the issue of delay.

In the query system, mapping entries between IDs (identifiers) and Locs (locators) are stored. Just like DNS, a query contain an ID could get a response include one or more Locs of this ID from the server in the query system.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

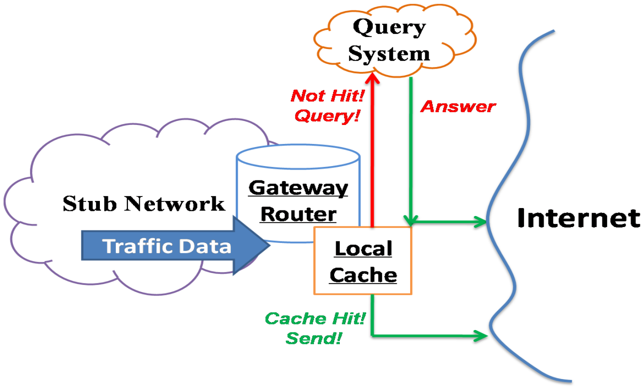

Fig. 1. Evaluation Model

B. The Gateway Router

The gateway router lies between a stub network and a backbone of the Internet. End host identifiers can be routed in stubs, but cannot be routed in the Internet backbones. For each packet with a certain destination identifier, a gateway router it passes should find out the corresponding locator and convert the packet into a version that can be routed within the backbone, via either the approach of encapsulation or translation (like what 6Core does [11]).

However, retrieving the mapping index service is a very time-consuming work. Therefore, any gateway couldn't do the retrieval for every packet. It must maintain a cache in a certain way. If the proper mapping entry is found in the cache, the packet can be immediately converted and transmitted into the backbones. If the router does not contain the corresponding entry, it has to send a retrieval to the query system. Until the answer is received, the packet should be queued, or be dropped if the queue length has exceeded the limit of the router.

In the canonical model, we define the local cache for mapping is working in the way of First-In-First-Out (FIFO) with priority tuning. A mapping entry just obtained from the query system will be put at the head of the cache, and be moved towards the end of the cache along with the time. However, if one entry in the middle of the cache is hit by a packet, the priority tuning mechanism works, treating the hit entry as it were just retrieved freshly and therefore moving it at the head again. Thus frequently used entries (corresponding to the most popular identifiers) will be kept in the cache. In such a way, we can expect that only a small part of packets will trigger long-latency retrieval over the query system, and accordingly the average performance of packet transmission will not be significantly degraded, due to the latency of retrieving mapping information from networks.

C. Metrics for Evaluation

For each component of the model, we concern certain metrics for its performance. For the local cache, the hit ratio in stable state represents the feasibility of the whole model. If really most of destination addresses in packets passing through a gateway router can be matched with certain entries in local cache of the mapping, it is not necessary to happen that a lot of queries and replies traverse over long-latency networks. Besides, bootstrapping behavior of the cache is also important. For this behavior, we concern the metric of the time duration from empty table to achieving the dynamic equilibrium.

Packets not matched with any cached entry raise queries over networks. The number of retrievals sent by the gateway router to the query system within a unit of time, and the actual queue length of the router, represents the performance of interaction between the local caching mechanism and the query system. Furthermore, the maximum query processed per second describes the capability of the query system itself.

III. Evaluations

A. Data Collection

Since the ID/Loc separation facilities have not been widely deployed, we just use the existing routing architecture instead: the IP prefixes in the Forwarding Information Base (FIB) of the Default Free Zone (DFZ) are used as routing locators and the IP addresses are used as end host identifiers. If one IP address belongs to a FIB routing entry, then the "identifier" can be mapped to the "locator". The FIB data we used are obtained from Route Views Project. The data set contains 261921 prefixes, so the query system mentioned above has 261921 mapping entries. Although the number of locators may be smaller than 200000 if ID-Loc separation has widely deployed, the number of Locs is difficult to predict. And the growth of the Internet and the large address space of IPv6 may bring more entries.

Because "identifier" and "locator" are chosen from today's Internet, real traffic data from the Internet could be used. We have gathered two groups of data from Internet for evaluation. One is from CERNET-TEIN2 outbound link with link speed of 250Mbps(Data Group No. 1). The other is from CERNET international outbound link with link speed of 6000Mbps(Data Group No. 2). The former includes the first 64 bytes of all the packets with time-stamps in a 24-hour period. The latter data is the destination address sequence in a continuous 24-hour period. In order to evaluate the performance of the query system using DNS, we have captured the first 96 bytes of all the DNS packets in the 24-hour period from the main DNS server of Tsinghua University Network, which has a peak link speed above 2000Mbps.

B. Performance Evaluation

According to the metrics of the model, the performance evaluation consists of three parts. At first, the evaluation begins with the hit ratio and bootstrapping stability of local cache for mapping entries. Then the evaluation focuses on the response time of the query system. Finally, the performance of the gateway router under given cache hit ratio and query response time will be shown.

1) Local Cache Performance: We use programs to simulate the model in Fig. 1. Traffic data are sent to the "gateway", the gateway router searches identifier-locator (ID/Loc) mapping entries in the local cache first. If a proper entry is found, the

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

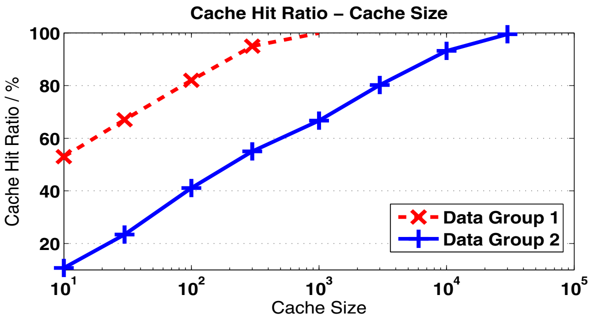

Fig. 2. Mapping Entry Cache Hit Ratio

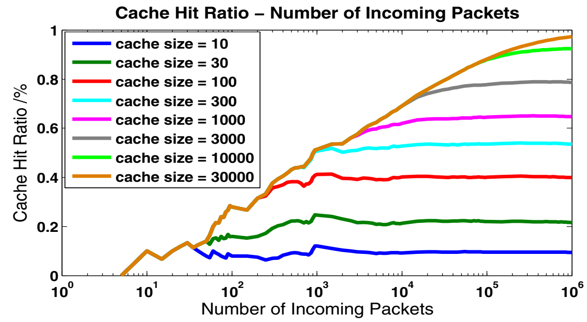

Fig. 4. Cache Hit Ratio Stability of Data Group 2

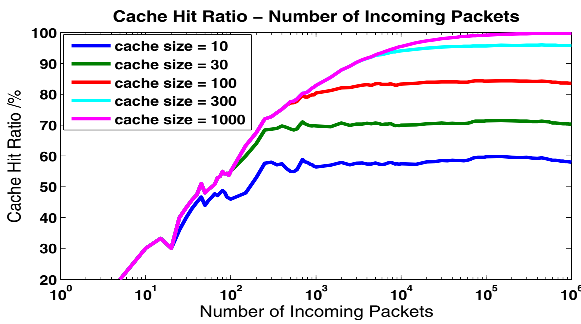

Fig. 3. Cache Hit Ratio Stability of Data Group 1

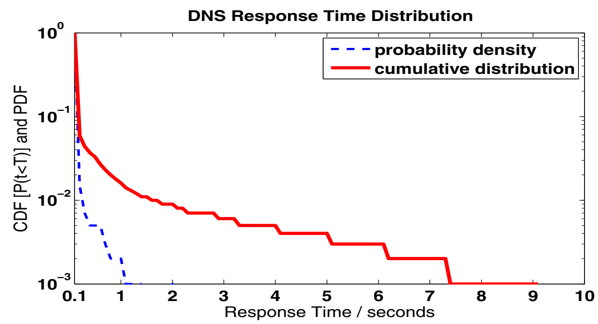

Fig. 5. DNS Response Time Distribution

cache is "hit" and the packet could be sent out immediately. If the router cannot find proper entries, the packet will be dropped or queued, and a query will be sent to the query system. We focus on hit ratio which represents the performance. Of course, the hit ratio depends on the size of the cache. After the feeding of 24-hour traffic data flow to the gateway, a final cache hit ratio could be calculated under given cache size. Fig. 2 shows the relationship between cache hit ratio and cache size. Logarithmic presentation is applied for ordinates. The higher curve is generated from our Data Group 1 while the lower is from Data Group 2.

In Data Group 1, the ratio reaches 99.9% when the cache size is 1000 entries, which is only 0.4% of the total number of locators. However, the ratio of Data Group 2 could reach 99% only when the cache size is above 30000 entries. On the one hand, the traffic of Data Group 2 is much heavier than Data Group 1. On the other hand, the traffic data of Data Group 2 may be more diverse than Data Group 1. Despite the differences, we could conclude that a small size of local cache can improve the performance a lot.

There may be two questions you have now in your mind, one is how stable the ratio is, the other is what will happen when the cache begins to bootstrap. We have run the program (of course, begin with empty cache) for only the first one million packets of each data group, and calculated the hit ratio when each packet arrived. The results are shown in Fig. 3 for Data Group 1 and Fig. 4 for Data Group 2.

According to the results, after the bootstrapping process, Hit ratios become stationary. The duration of the process depends on the cache size. In our data, one million packets is large enough for a 99%-ratio-size cache completing the bootstrap.

2) Query System Performance: The performance of the query system could be measured by two variables: response time and maximum queries per second. Using our 24-hour period DNS traffic data, response time could be calculated. For each query packet, the program will find the answer packet for it and then use time stamps of both packets to calculate the response time using simple substraction. It needs to be point out that our data were captured on the DNS server, so the response time will not contain the time consumed during transport. In order to get the best performance, the link between the query system and the router should be faster, so we will not consider the transport time in the evaluation. Otherwise, the average Round Trip Time (RTT) could be added for evaluation.

The probability density function and cumulative distribution function of the response time are shown in Fig. 5. The average response time of all the queries is 68ms. Because of the cache mechanism in DNS server, most of the response time is shorter than 100 milliseconds. Only 1% of all the response time data is longer than 1.5s. This gives the typical performance of the query system. However, the cache configuration may differ in different DNS servers. So we have to consider the response time of those no-cache entries. For the reason that the response time of a cached entry is usually below 200ms, we calculate the probability density data and cumulative distribution data again but exclude those with the response time smaller than 200ms. Results are shown in Fig. 6.

Unlike Fig. 5, the ordinates in Fig. 6 are linear. In this figure we can see only 10% queries have response time longer than

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

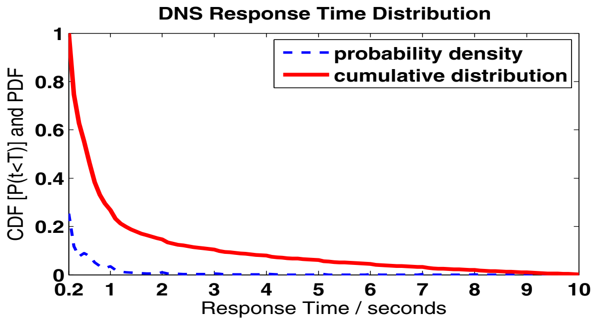

Fig. 6. DNS Response Time Distribution (except cached)

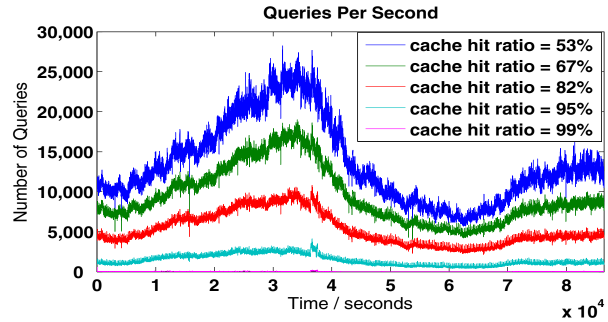

Fig. 7. Queries Per Second In 24 Hours

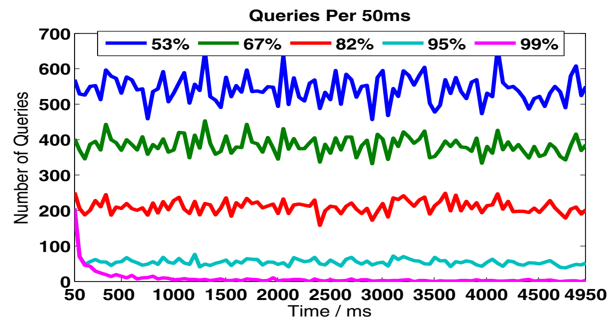

Fig. 8. Queries Per 50ms In 5s

2s, in all the queries which need recursive queries done by the DNS server. That means, if the gateway router choose to queue the packet when it does not have the proper entries, the packet could be sent in 1s in most cases. Even if the DNS server have a very small cache, most packets could be sent in 2s. These time data give great reference value for evaluating the performance of the gateway router. By the way, according to the data, our DNS server could handle at least 2000 queries per second. This data is useful although we don't know the maximum query per second.

3) Gateway Router Querying and Queueing Performance: Because the limitation of our data, only Data Group 2 has the time stamps for evaluation. Therefore we only use data of Data Group 2 for router side evaluation. Different traffic flow characteristics and different cache sizes lead to different cache hit ratios. Further more, different cache hit ratios may result in different number of dropped packets or different queue lengthes. Under the principle of best effort, the router should queue the packet first, and begin to drop packets when the queue is already full. How will the queue become full? It is a dynamic process. We can begin the analysis with extreme cases. If the router receives M packets which don't have proper cache entries in one second, those M packets would be queued, and at least M queries would be sent to the query system. In the next second, other M packets would come. So if the query system could return all the answers of the last second, the M packets in the queue could be sent out, and the M packets in the next second could be queued. But if the query system could not handle M queries per second, disaster will happen. The queue will keep growing – it will be full no matter how big the queue capacity is. Through the analysis above, we can see the maximum query the query system can handle per second is a limitation to the gateway router. Under this consideration, we have calculated the queries per second data of the gateway router under different cache hit ratios, shown as Fig. 7.

In Fig. 7 we can see, queries per second could reach 10,000 when the cache hit ratio is under 82%. For our query system evaluated in the former subsection, no packets will be dropped when the cache hit ratio is above 95% and the queue length is long enough. How about the queue length? The average response time of our query system is 68ms. So we have calculated the queries per 50ms in the first 5 seconds of the traffic data. The graph is depicted by Fig. 8.

The first point we can see in Fig. 8, the queries basically remain stable in small timescale (compare with 24 hours). For the hit ratio of 95%, there are about 50 queries every 50ms. If all the queries could be answered in 50ms, the queue of the gateway router could be kept as 50 packets. It is not very large, but remember it's only an average result, and the Round Trip Time (RTT) has not been taken into consideration yet. But we can still see if the local cache, the query system and the queue cooperate well, the performance could be satisfying.

The second point we should take notice of in Fig. 8 is bootstrapping. The left parts of the curves show us the higher number of queries in bootstrap process. Because of the non-established cache, bootstrap could be a problem unless the router has enough memory to queue the packets in bootstrap process.

IV. Discussions

A. Considerations on Performance

According to the evaluation, the first bottleneck of the performance is the maximum query that query system can handle in a time unit. If the queries sent by the router per time unit is more than that maximum value, the queue will keep growing all the time and become full finally. Then many packets will be dropped. To avoid this bottleneck, better cache mechanisms and larger cache should be used on the router side, and the query system should be better designed and deployed.

If the problem above could be avoided, a large enough memory queue in the router is able to solve all the problems including bootstrap problem. For the reason that the query system is not bottleneck now, the ceiling length of the queue could be estimated using the maximum queries that the query system can handle in a time unit with the response time for the queries. Given T as response time and Q as maximum query per time unit, the length of the queue could be $T*Q$. Normally, many of the queued packets should be the first packets of TCP sessions, so the bytes number could not be very large.

B. Modifications On End Hosts

For an end host, if it tries to access another host whose identifier-locator (ID/Loc) mapping entry is not in the router's cache, the packet will be queued, even dropped. The end host will have poor performance on this connection. If the packet is dropped, according to TCP requirements [12], the retransmission timeout of the first packet is around 3 seconds. Actually, the response time of the query system are always smaller than 1 second. When today's applications like Microsoft Internet Explorer, Mozilla Firefox Web Browser and many web acceleration programs are using build-in DNS cache to improve user experiences, why not make a slight modification on end hosts to save the users' time? There are at least two ways to do this modification.

The first way of the modification is just modifying the retransmission timeout and the timeout calculation algorithm. Let the first packet be a trigger. Begin to send data after the cache is ready.

The second way needs the participation of the router. The router could send back a ready information when the cache is ready. The identifier of the source host could be contained in the message sent to the query system so that the router does not need to keep states – when the router receives an answer, it could send ready message to the identifier contained in the answer packet, just like DNS server does.

At last, for DNS which uses UDP to send message, if the only packet is dropped, the performance becomes poor. We suggest those servers build in the trigger mechanism to improve performance.

V. Related Works

Iannone and Bonaventure have done a related work on performance evaluation for caching of the mapping entries through another point of view [13]. In their model from LISP, an unlimited-size local cache was included. In contrast, our model focuses the on the query-cache-queue cooperation, and has the assumption that the cache size may be limited by hardware or cost. We give conclusions under limited cache size, which complemented their works.

VI. Conclusion

We try to answer a question: if we split the roles of Internet addresses, is it possible that we also split the works for the mapping and forwarding in the Internet infrastructure? Some approaches argue this would be very harsh, but our works try to make a quantitative evaluation based on current existing data and methodologies.

For this purpose, we have proposed a canonical model for the mapping facility that contains a query resolver, a local mapping cache and a queueing machine in a gateway router, and applying real traffic data into the model to evaluate the impact of deploying the mapping service rather than integrating the mapping with routing infrastructure. Simulation results suggest that the mapping facility can work with end-to-end performance impacted by the settings in caching and the mapping service, which are hardly limited by hardware and deployment costs. Fortunately, however, we can overcome this impact to some extent by modifying TCP behaviors and parameters. According to the evaluation work in this paper, we conclude that applying a independent service facility for ID/Loc mapping is feasible.

People have committed that the Internet structure must be changed when it grows too much in scale [9], and the same philosophy is applicable for the end systems — once the overall structure of the Internet has changed, it is not difficult to understand that end system behaviors must be changed as well. How do we make end system completely accommodate the new Internet architecture where identifier and locator are split? This will be an important question to be answered in the ongoing works.

References

[1] D. Meyer, L. Zhang, and K. Fall, "RFC 4984: Report from the iab workshop on routing and addressing," Sep. 2007, status: INFORMATIONAL. [Online]. Available: http://www.ietf.org/rfc/rfc4984.txt

[2] B. Quoitin, L. Iannone, C. de Launois, and O. Bonaventure, "Evaluating the benefits of the locator/identifier separation," in *Proceedings of ACM SIGCOMM 2007 Workshops*, Kyoto, Japan, August 2007, pp. 35–40.

[3] D. Farinacci, V. Fuller, D. Oran, and D. Meyer, "Locator/ID Separation Protocol (LISP)," draft-farinacci-lisp-06.txt, Feb. 2008, work in progress. [Online]. Available: http://tools.ietf.org/html/draft-farinacci-lisp-06

[4] D. Farinacci, D. Oran, V. Fuller, and J. Schiller, "Locator/ID Separation Protocol (LISP2)[DNS-based Version]," lisp2.ppt, Nov. 2006, work in progress. [Online]. Available: http://www.dinof.net/ dino/ietf/lisp2.ppt

[5] S. Brim, N. Chiappa, D. Farinacci, V. Fuller, D. Lewis, and D. Meyer, "LISP-CONS: A Content distribution Overlay Network Service for LISP," draft-meyer-lisp-cons-03.txt, Nov. 2007, work in progress. [Online]. Available: http://tools.ietf.org/html/draft-meyer-lisp-cons-03

[6] D. Farinacci, V. Fuller, and D. Meyer, "LISP Alternative Topology (LISP-ALT)," draft-fuller-lisp-alt-01.txt, Nov. 2007, work in progress. [Online]. Available: http://tools.ietf.org/html/draft-fuller-lisp-alt-01

[7] E. Lear, "NERD: A Not-so-novel EID to RLOC Database," draft-lear-lisp-nerd-03.txt, Jan. 2008, work in progress. [Online]. Available: http://tools.ietf.org/html/draft-lear-lisp-nerd-03

[8] D. Jen, M. Meisel, D. Massey, L. Wang, B. Zhang, and L. Zhang, "APT: A Practical Transit Mapping Service," draft-jen-apt-01.txt, Nov. 2007, work in progress. [Online]. Available: http://tools.ietf.org/html/draft-jen-apt-01

[9] D. Massey, L. Wang, B. Zhang, and L. Zhang, "A scalable routing system design for future internet," in *Proceedings of ACM SIGCOMM 2007 Workshops*, Kyoto, Japan, August 2007, pp. 403–408.

[10] C. Huitema, "RFC 1383: An experiment in DNS based IP routing," Dec. 1992, status: EXPERIMENTAL. [Online]. Available: http://www.ietf.org/rfc/rfc1383.txt

[11] H. Zhang and M. Chen, "Forming an ipv6-only core for todays internet," in *Proceedings of ACM SIGCOMM 2007 Workshops*, Kyoto, Japan, August 2007, pp. 379–384.

[12] R. T. Braden, "RFC 1122: Requirements for Internet hosts — communication layers," Oct. 1989, status: STANDARD. [Online]. Available: http://www.ietf.org/rfc/rfc1122.txt

[13] L. Iannone and O. Bonaventure, "On the cost of caching locator/id mappings." 3rd Annual CoNEXT Conference, December 2007.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.